AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently amended) A device operating as a finite state machine and provided for processing events and actions relating to at least one object ~~to be moved~~ that changes between an initial and a final state, said device comprising:

~~a processing member connected to a memory, wherein said initial and final state are integrated into at least one event-state-action diagram defining said finite state machine, said memory having an input connected to a state engine editor and being provided for storing said at least one diagram, said~~

a state engine editor having an input for receiving object data[[,]] identifying said object ~~and said initial and final state~~, said object data further comprising data relating to:

a first set of states including the initial state and the final state,

a second set of events, and

a third set of actions, said third set comprising data relating to at least one of a first sub-set of processing actions, a second sub-set of timed actions ~~provided to initiate~~, or a third sub-set of transition actions, at least one of the timed actions being initiated each time at least one ~~predetermined action~~ of the processing actions of said first sub-set lapses after a predetermined time period ~~has lapsed and a third sub-set of transition actions,~~;

said state engine editor being ~~provided for forming said~~ configured to:

form at least one event-state-action diagram defining the finite state machine, the at least one diagram being structured as a matrix of said states of said first set and said events of said second set in order to create at least one state-event combination in at least one position ~~, at positions~~ within said matrix, and ~~a state-event combination, said state engine editor being further provided for attributing to~~

associate at least one of said actions of said third set to the at least one state-event combination; ~~at least one of said actions of said third set,~~

a memory having an input connected to the state engine editor and being configured to store the at least one diagram;

a processing member connected to the memory, said processing member having an input ~~for receiving event-messages~~ and being ~~provided for converting~~ configured to:

receive at least one event-message via the input of the processing member;

determine which of the events of the second set is associated with the at least one [[a]] received event-message, ~~into one of said events of said second set, said processing member being also provided for monitoring said states in order to recognize~~

monitor an actual state [[for]] of said object, the actual state being associated with one of the states of the first set, ~~and for selecting within said state diagram, upon receipt of one of said event-messages,~~

determine a position within the matrix of said diagram corresponding to said actual state and said determined event, and ~~obtained by said converting, said~~

~~processing member being further provided for retrieving said actions located at said selected~~

determine at least one action associated with the determined position; and ~~for supplying said retrieved actions to an action dispatcher in order to execute said retrieved action, said~~

an action dispatcher ~~comprising for each action of said third set~~ configured to receive the at least one determined action from the processing member, and perform an execution routine ~~provided for controlling said execution of said~~ to execute the at least one determined action.

2. (Currently amended) A device as claimed in claim 1, wherein said state engine editor is ~~provided for attributing to~~ configured to associate with each of the at least one state-event combination~~, comprising said initial state,~~ an action selected from among said third sub-set.

3. (Currently amended) A device as claimed in claim 1 wherein said state engine editor is ~~provided for attributing to~~ configured to associate with each of the at least one state-event combination~~, comprising said final state,~~ only actions belonging to said first or second sub-set.

4. (Canceled)

5. (Canceled)

6. (Currently amended) A device as claimed in claim 1, wherein each diagram is identified in the memory by a description thereof and a reference to the object to which [[it]] the respective diagram belongs.

7. (Currently amended) A device as claimed in claim 1, wherein:

the data relating to each state ~~is identified by~~ includes a description thereof and a reference to the diagram to which [[it]] the respective state belongs; and

the data relating to each event includes a description thereof and a reference to the object to which the respective event belongs.

8. (Canceled)

9. (Canceled)

10. (Currently amended) A device as claimed in claim 1, wherein said state engine editor is ~~provided for forming said diagrams~~ configured to form the at least one diagram with an XML description.

11. (Currently amended) A device as claimed in claim 1, wherein the data relating to each transition action of said third sub-set comprises a reference to an event of said second set, a source and a target state as well as a reference to said diagram to which [[it]] the respective transition action belongs.

12. (Currently amended) A device as claimed in claim 1, wherein said first sub-set comprises a first class of generic actions and a second class of specific actions, and the data relating to each action of the first sub-set includes ~~identified by~~ a reference to the object to which [[it]] the respective action belongs.

13. (Currently amended) A method for processing, within a finite state machine, events and actions relating to at least one object ~~to be moved~~ that changes between an initial and a final state, wherein said method comprises:

receiving object data identifying said at least one object and said initial and final state, said object data comprising a first set of states, a second set of events and a third set of actions, said third set comprising a first sub-set of processing actions, a second sub-set of timed actions provided to initiate each time at least one predetermined action of said first sub-set has lapsed after a predetermined time period ~~has lapsed~~ and a third sub-set of transition actions;

forming at least one event-state-action diagram defined in a final state machine pattern of said finite state machine by structuring said states of said first set and said events of said second set as a matrix in order to create at least one state-event combination in at least one position ~~at positions~~ within said matrix, ~~each time, a state-event combination~~ the diagram including the initial and final states;

~~integrating said initial and final state into said diagram;~~

~~attributing to~~ associating with at least one state-event combination at least one of said actions of said third set;

receiving at least one event-message ~~event-messages~~ relating to said at least one object;

~~and converting a~~ determining which of the events of the second set is associated with the at least one received event-message ~~into one of said events of said second set~~;

~~recognizing~~ determining said at least one object ~~into~~ associated with said at least one received event-message;

monitoring ~~said states in order to recognize~~ an actual state for said ~~recognized~~ determined object; ~~and~~

~~selecting within said state diagram~~ determining, upon receipt of ~~one of said event-messages~~ the at least one event-message, a position within the matrix of said diagram corresponding to said actual state and said determined event ~~obtained by said converting~~;

~~retrieving said actions located at said selected~~ determining at least one action associated with the determined position; and

executing said ~~retrieved~~ determined action by processing an execution routine provided for controlling said execution of said ~~retrieved~~ determined action.

14. (Currently amended) A method as claimed in claim 13, wherein:

said at least one object includes ~~plural~~ a plurality of objects~~, said objects~~ classified by object types, ~~each object being identified by~~ the object data identifying the plurality of objects including a definition and a description of the object type to which [[it]] the respective object belongs, and

~~wherein~~ a plurality of event-state-action diagrams are formed for each ~~object~~ of the objects, each ~~diagram~~ of the diagrams corresponding to one of said object types, said monitoring further comprising a selection of at least one of said diagrams based on said object type.

15. (Canceled)

16. (Currently amended) A method as claimed in claim 13, wherein said at least one object includes a plurality of objects, and for each of the plurality of objects, at least one dedicated event-state-action diagram is formed, ~~said event-messages~~ the at least one event-message comprising an object identifier, ~~said monitoring~~ the method further comprising ~~a selection of~~ selecting at least one of said diagrams based on said ~~identified~~ object identifier from the received event-message.

17. - 20. (Canceled)

21. (New) The system of claim 21, wherein the processing member determines the at least one action associated with the determined position independent of information from any other position in the matrix.

22. (New) A system for monitoring and controlling delivery of a package, the system comprising:

a memory configured to store package data identifying the package, the package data comprising data relating to a first set of states including an initial state and a final state of the package during a package delivery, a second set of events, and a third set of actions, the package data incorporated into an event-state-action diagram, the diagram being structured as a matrix of the states of the first set and the events of the second set to form a plurality of state-event combinations, the diagram associating at least one of the actions of the third set to at least one of the state-event combinations;

a processing member connected to the memory, the processing member having an input and being configured to:

receive at least one event-message via the input of the processing member,

determine which of the events of the second set is associated with the at least one received event-message,

monitor an actual state of the package, the actual state being associated with one of the states of the first set, and

determine, using the stored diagram and based on the determined event and the actual state of the package, which of the actions of the third set to perform; and

an action dispatcher configured to receive the at least one determined action from the processing member and perform an execution routine to execute the at least one determined action.

23. (New) The system of claim 22, wherein the event-message is received from a shipping application used in a warehouse, a scanner used in the warehouse, a scanner used by a courier driver, a scanner used at a drop-off location, or a timer trigger.

24. (New) The system of claim 22, wherein the initial state relates to a source of the package, and the final state relates to a final destination for the package.

25. (New) The system of claim 22, wherein the actual state of the package is determined independently from data stored in the diagram.

26. (New) The system of claim 22, wherein the action dispatcher is capable of performing different actions to transition the package from a first state to a second state.

27. (New) The system of claim 26, wherein the processing member is configured to select, based on the actual state of the package and the determined event, which of the different actions to perform to transition the package from the first state to the second state.

28. (New) The system of claim 22, wherein, when the actual state of the package is a first state, the action dispatcher is capable of performing different actions to either transition the package to a second state or to a third state.

29. (New) The system of claim 28, wherein the processing member is configured to select, based on the actual state of the package and the determined event, which of the different actions to perform to transition the package from the first state to either the second state or the third state.